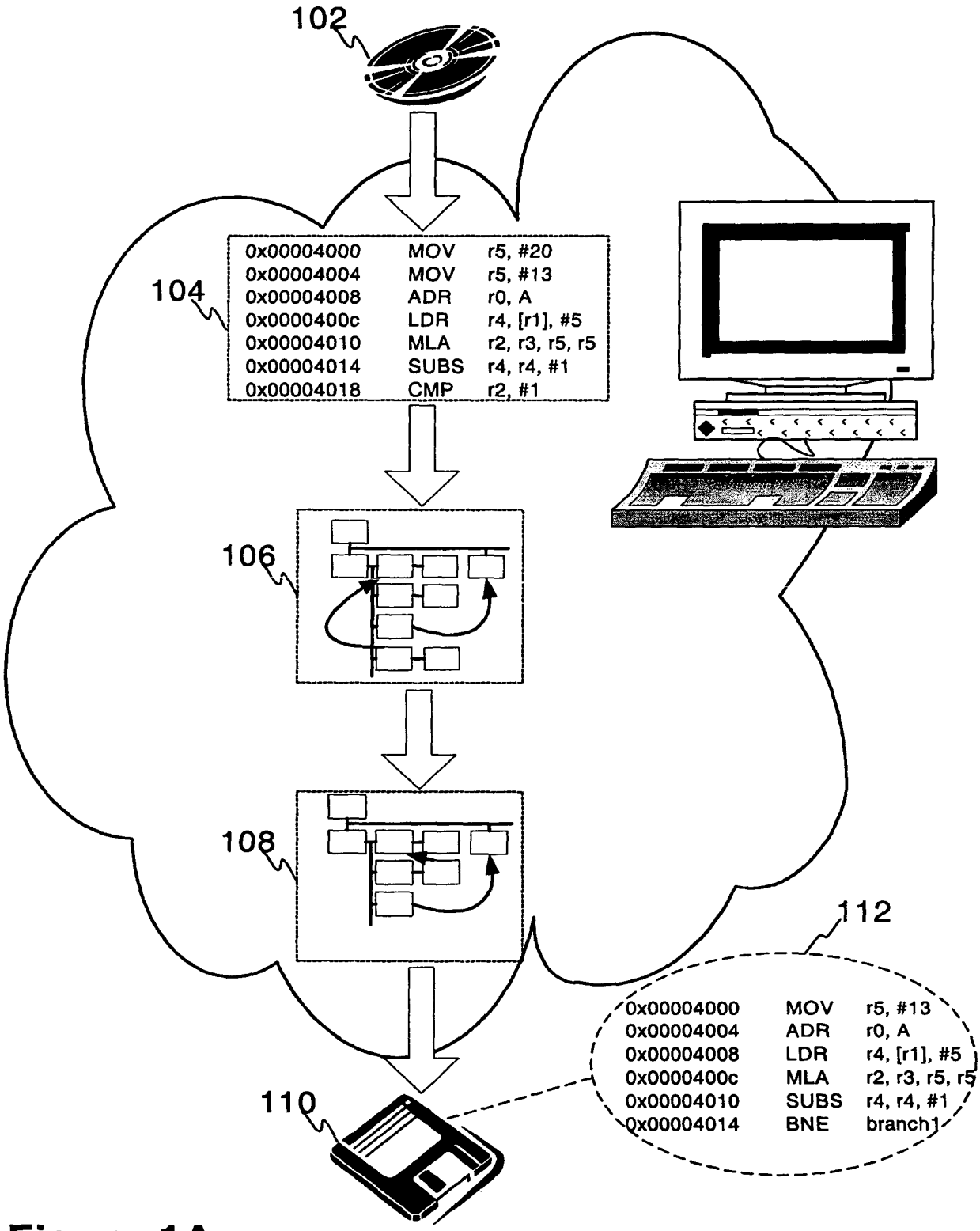

102
104
0x00004000 MOV r5, #20
0x00004004 MOV r5, #13
0x00004008 ADR r0, A
0x0000400c LDR r4, [r1], #5
0x00004010 MLA r2, r3, r5, r5
0x00004014 SUBS r4, r4, #1
0x00004018 CMP r2, #1
106
108
112
0x00004000 MOV r5, #13
0x00004004 ADR r0, A
0x00004008 LDR r4, [r1], #5
0x0000400c MLA r2, r3, r5, r5
0x00004010 SUBS r4, r4, #1
0x00004014 BNE branch1
110

Figur 1A

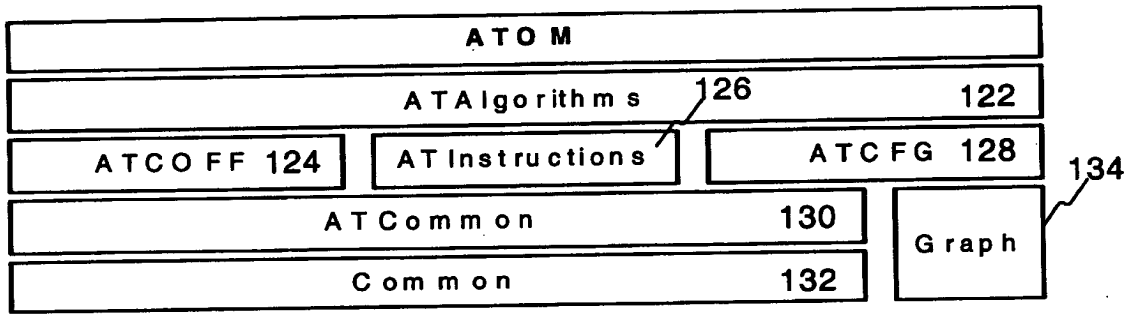
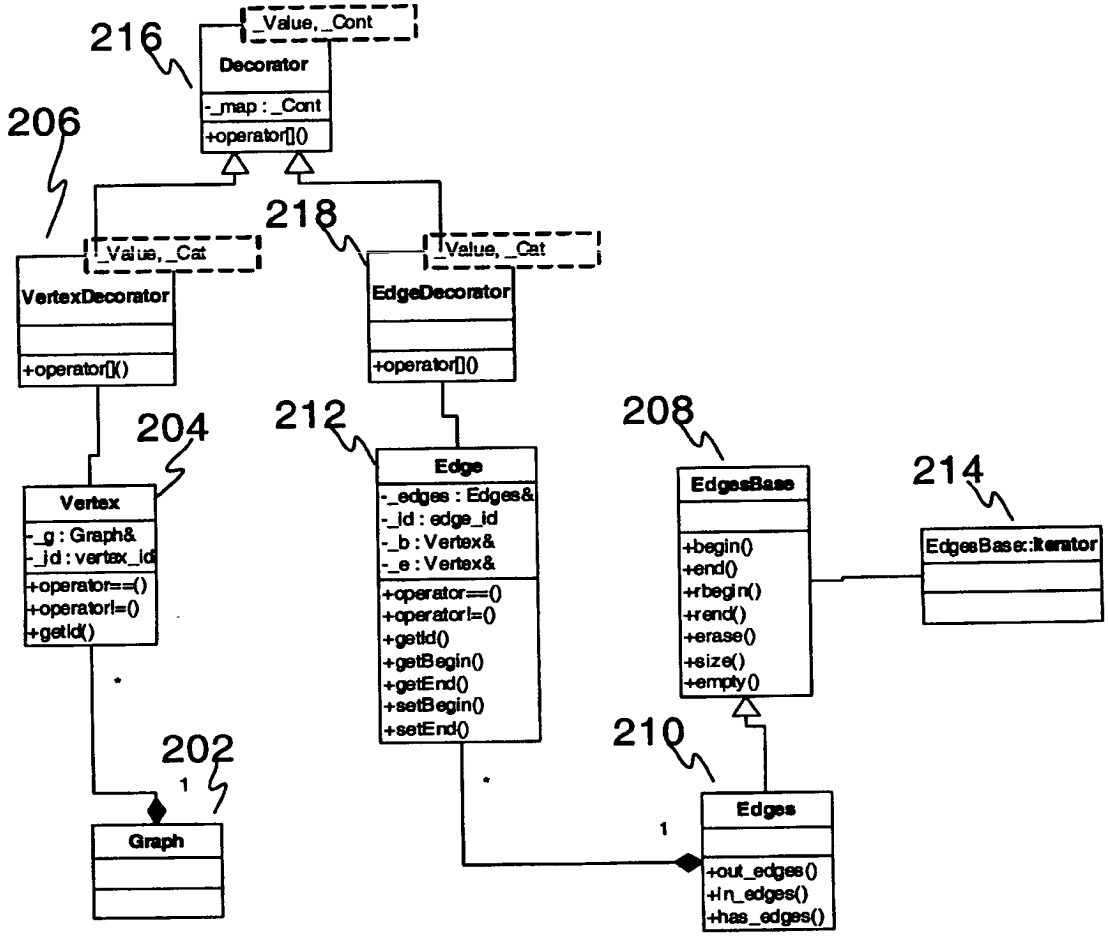


Figure 1B



Figur 2

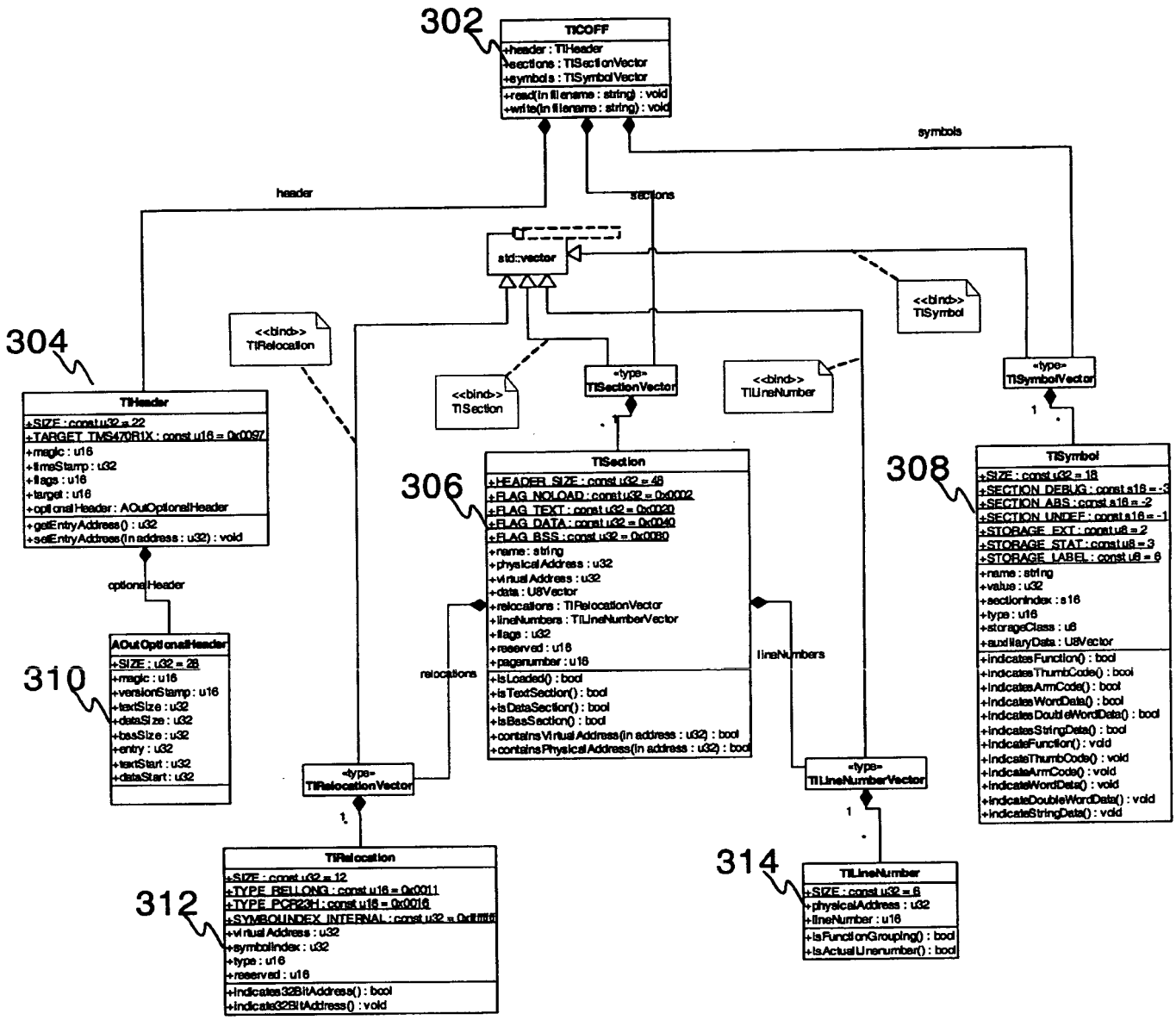

302
TICOFF
+header : TIHeader
+sections : TISectionVector
+symbols : TISymbolVector
+read(in filename : string) : void
+write(in filename : string) : void
symbols
header
sections
std::vector
<<bind>>
TISymbol
304
<<bind>>
TIRelocation
<<bind>>
TISection
«type»
TISectionVector
<<bind>>
TILineNumber
«type»
TISymbolVector
1
TIHeader
+SIZE : const u32 = 22
+TARGET_TMS470R1X : const u16 = 0x0097
+magic : u16
+timeStamp : u32
+flags : u16
+target : u16
+optionalHeader : AOutOptionalHeader
+getEntryAddress() : u32
+setEntryAddress(in address : u32) : void
306
TISection
+HEADER_SIZE : const u32 = 48
+FLAG_NOLOAD : const u32 = 0x0002
+FLAG_TEXT : const u32 = 0x0020
+FLAG_DATA : const u32 = 0x0040
+FLAG_BSS : const u32 = 0x0080
+name : string
+physicalAddress : u32
+virtualAddress : u32
+data : U8Vector
+relocations : TIRelocationVector
+lineNumbers : TILineNumberVector
+flags : u32
+reserved : u16
+pagenumber : u16
+isLoaded() : bool
+isTextSection() : bool
+isDataSection() : bool
+isBssSection() : bool
+containsVirtualAddress(in address : u32) : bool
+containsPhysicalAddress(in address : u32) : bool
308
TISymbol
+SIZE : const u32 = 18
+SECTION_DEBUG : const s16 = -3
+SECTION_ABS : const s16 = -2
+SECTION_UNDEF : const s16 = -1
+STORAGE_EXT : const u8 = 2
+STORAGE_STAT : const u8 = 3
+STORAGE_LABEL : const u8 = 6
+name : string
+value : u32
+sectionIndex : s16
+type : u16
+storageClass : u8
+auxiliaryData : U8Vector
+indicatesFunction() : bool
+indicatesThumbCode() : bool
+indicatesArmCode() : bool
+indicatesWordData() : bool
+indicatesDoubleWordData() : bool
+indicatesStringData() : bool
+indicateFunction() : void
+indicateThumbCode() : void
+indicateArmCode() : void
+indicateWordData() : void
+indicateDoubleWordData() : void
+indicateStringData() : void
optionalHeader
310
AOutOptionalHeader
+SIZE : u32 = 28
+magic : u16
+versionStamp : u16
+textSize : u32
+dataSize : u32
+bssSize : u32
+entry : u32
+textStart : u32
+dataStart : u32
relocations
lineNumbers
«type»
TIRelocationVector
«type»
TILineNumberVector
1
1
312
TIRelocation
+SIZE : const u32 = 12
+TYPE_RELLONG : const u16 = 0x0011
+TYPE_PCR23H : const u16 = 0x0016
+SYMBOLINDEX_INTERNAL : const u32 = 0xffffff
+virtualAddress : u32
+symbolIndex : u32
+type : u16
+reserved : u16
+indicates32BitAddress() : bool
+indicate32BitAddress() : void
314
TILineNumber
+SIZE : const u32 = 6
+physicalAddress : u32
+lineNumber : u16
+isFunctionGrouping() : bool
+isActualLinenumber() : bool

Figure 3

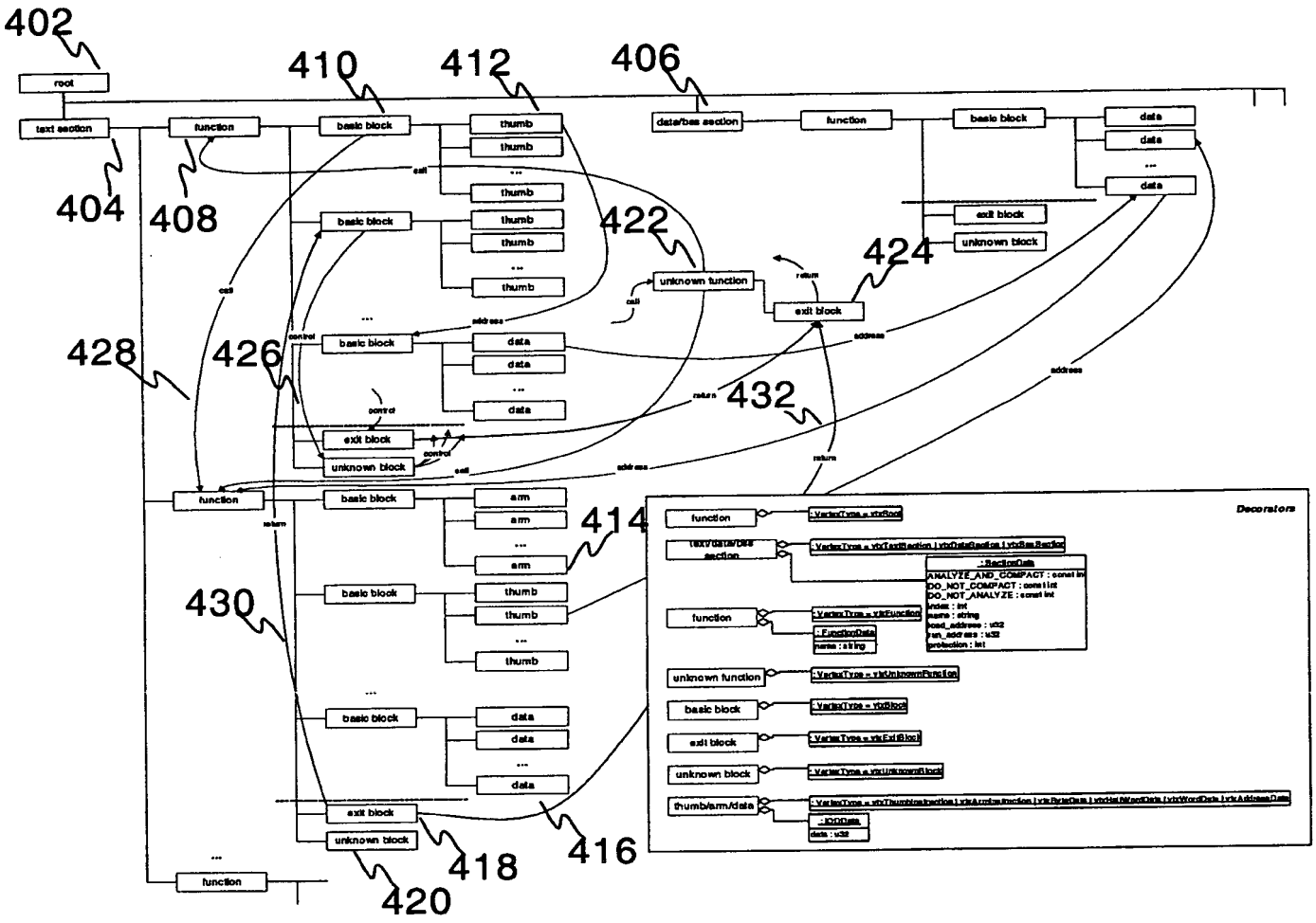


Figure 4

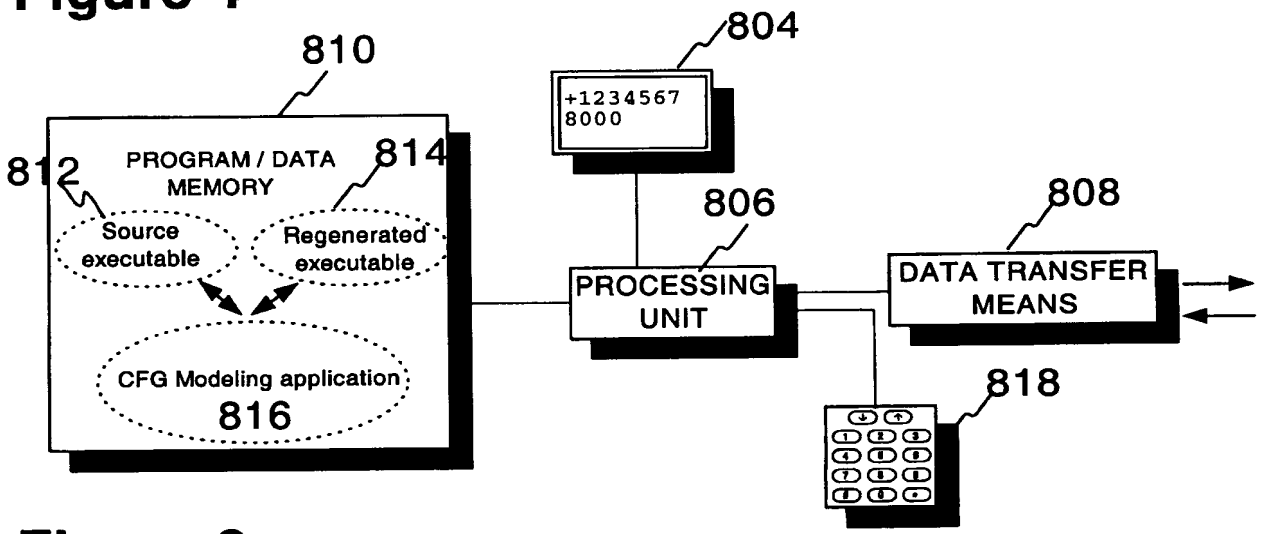


Figure 8

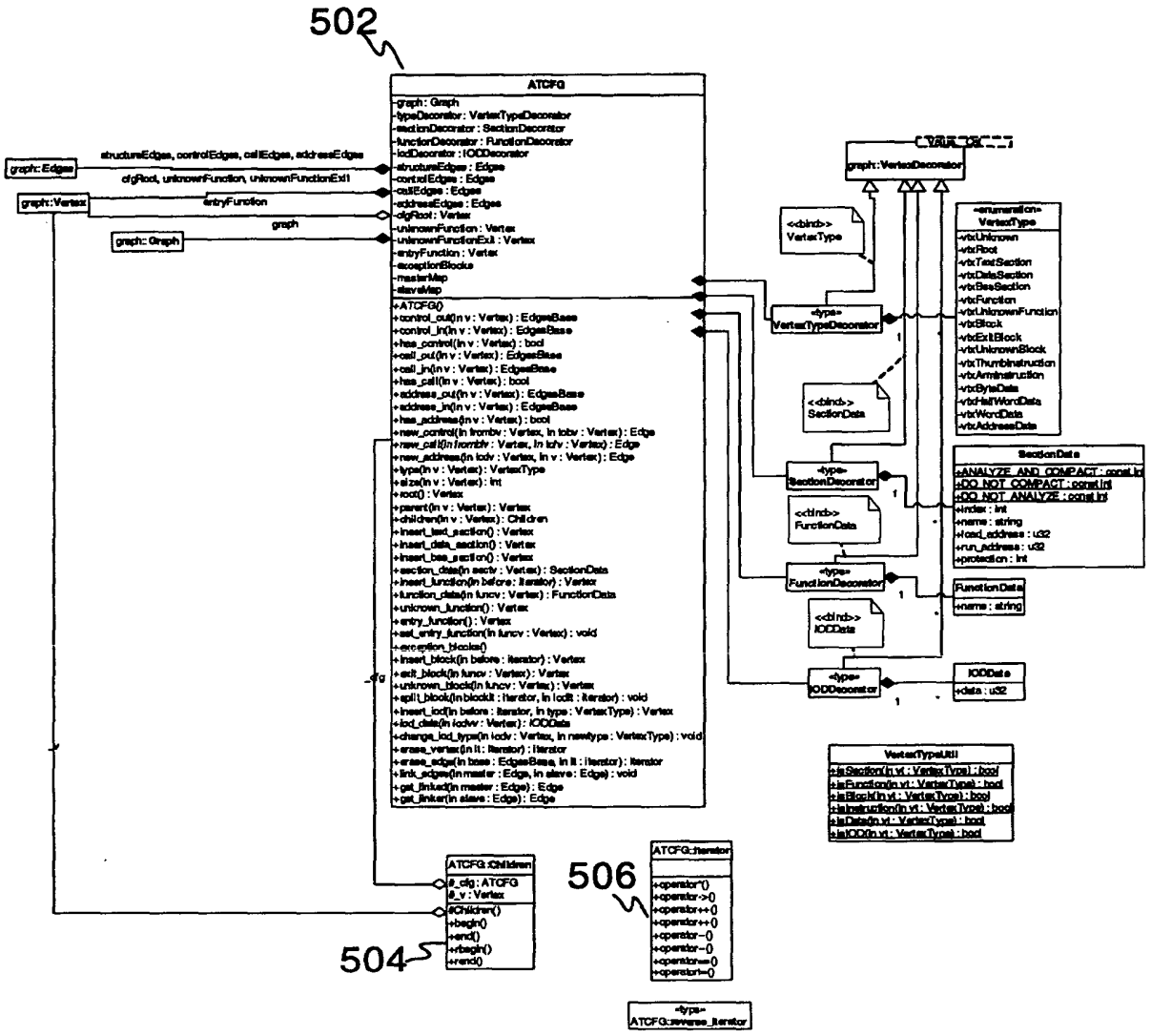

502
ATCFG
-graph : Graph
-typeDecorator : VertexTypeDecorator
-sectionDecorator : SectionDecorator
-functionDecorator : FunctionDecorator
-iodDecorator : IODDecorator
-structureEdges : Edges
-controlEdges : Edges
-callEdges : Edges
-addressEdges : Edges
-cfgRoot : Vertex
-unknownFunction : Vertex
-unknownFunctionExit : Vertex
-entryFunction : Vertex
-exceptionBlocks
-masterMap
-slaveMap
+ATCFG()
+control_out(in v : Vertex) : EdgesBase
+control_in(in v : Vertex) : EdgesBase
+has_control(in v : Vertex) : bool
+call_out(in v : Vertex) : EdgesBase
+call_in(in v : Vertex) : EdgesBase
+has_call(in v : Vertex) : bool
+address_out(in v : Vertex) : EdgesBase
+address_in(in v : Vertex) : EdgesBase
+has_address(in v : Vertex) : bool
+new_control(in fromv : Vertex, in tov : Vertex) : Edge
+new_call(in fromfv : Vertex, in tofv : Vertex) : Edge
+new_address(in iodv : Vertex, in v : Vertex) : Edge
+type(in v : Vertex) : VertexType
+size(in v : Vertex) : int
+root() : Vertex
+parent(in v : Vertex) : Vertex
+children(in v : Vertex) : Children
+insert_text_section() : Vertex
+insert_data_section() : Vertex
+insert_bss_section() : Vertex
+section_data(in sectv : Vertex) : SectionData
+insert_function(in before : Iterator) : Vertex
+function_data(in funcv : Vertex) : FunctionData
+unknown_function() : Vertex
+entry_function() : Vertex
+set_entry_function(in funcv : Vertex) : void
+exception_blocks()
+insert_block(in before : Iterator) : Vertex
+exit_block(in funcv : Vertex) : Vertex
+unknown_block(in funcv : Vertex) : Vertex
+split_block(in blockIt : Iterator, in iodIt : Iterator) : void
+insert_iod(in before : Iterator, in type : VertexType) : Vertex
+iod_data(in iodvv : Vertex) : IODData
+change_iod_type(in iodv : Vertex, in newtype : VertexType) : void
+erase_vertex(in It : Iterator) : Iterator
+erase_edge(in base : EdgesBase, in It : Iterator) : Iterator
+link_edges(in master : Edge, in slave : Edge) : void
+get_linked(in master : Edge) : Edge
+get_linker(in slave : Edge) : Edge
graph: Edges
structureEdges, controlEdges, callEdges, addressEdges
cfgRoot, unknownFunction, unknownFunctionExit
graph:Vertex
entryFunction
graph
graph: Graph
cfg
graph:VertexDecorator
<<bind>>
VertexType
«enumeration»
VertexType
-vtxUnknown
-vtxRoot
-vtxTextSection
-vtxDataSection
-vtxBssSection
-vtxFunction
-vtxUnknownFunction
-vtxBlock
-vtxExitBlock
-vtxUnknownBlock
-vtxThumbInstruction
-vtxArmInstruction
-vtxByteData
-vtxHalfWordData
-vtxWordData
-vtxAddressData
«type»
VertexTypeDecorator
1
<<bind>>
SectionData
«type»
SectionDecorator
1
SectionData
+ANALYZE_AND_COMPACT : const int
+DO_NOT_COMPACT : const int
+DO_NOT_ANALYZE : const int
+index : int
+name : string
+load_address : u32
+run_address : u32
+protection : int
<<bind>>
FunctionData
«type»
FunctionDecorator
1
FunctionData
+name : string
<<bind>>
IODData
«type»
IODDecorator
1
IODData
+data : u32
VertexTypeUtil
+isSection(in vt : VertexType) : bool
+isFunction(in vt : VertexType) : bool
+isBlock(in vt : VertexType) : bool
+isInstruction(in vt : VertexType) : bool
+isData(in vt : VertexType) : bool
+isIOD(in vt : VertexType) : bool
ATCFG::Children
#_cfg : ATCFG
#_v : Vertex
#Children()
+begin()
+end()
+rbegin()
+rend()
504
506
ATCFG::Iterator
+operator*()
+operator->()
+operator++()
+operator++()
+operator--()
+operator--()
+operator==()
+operator!=()
«type»
ATCFG::reverse_Iterator

Figure 5

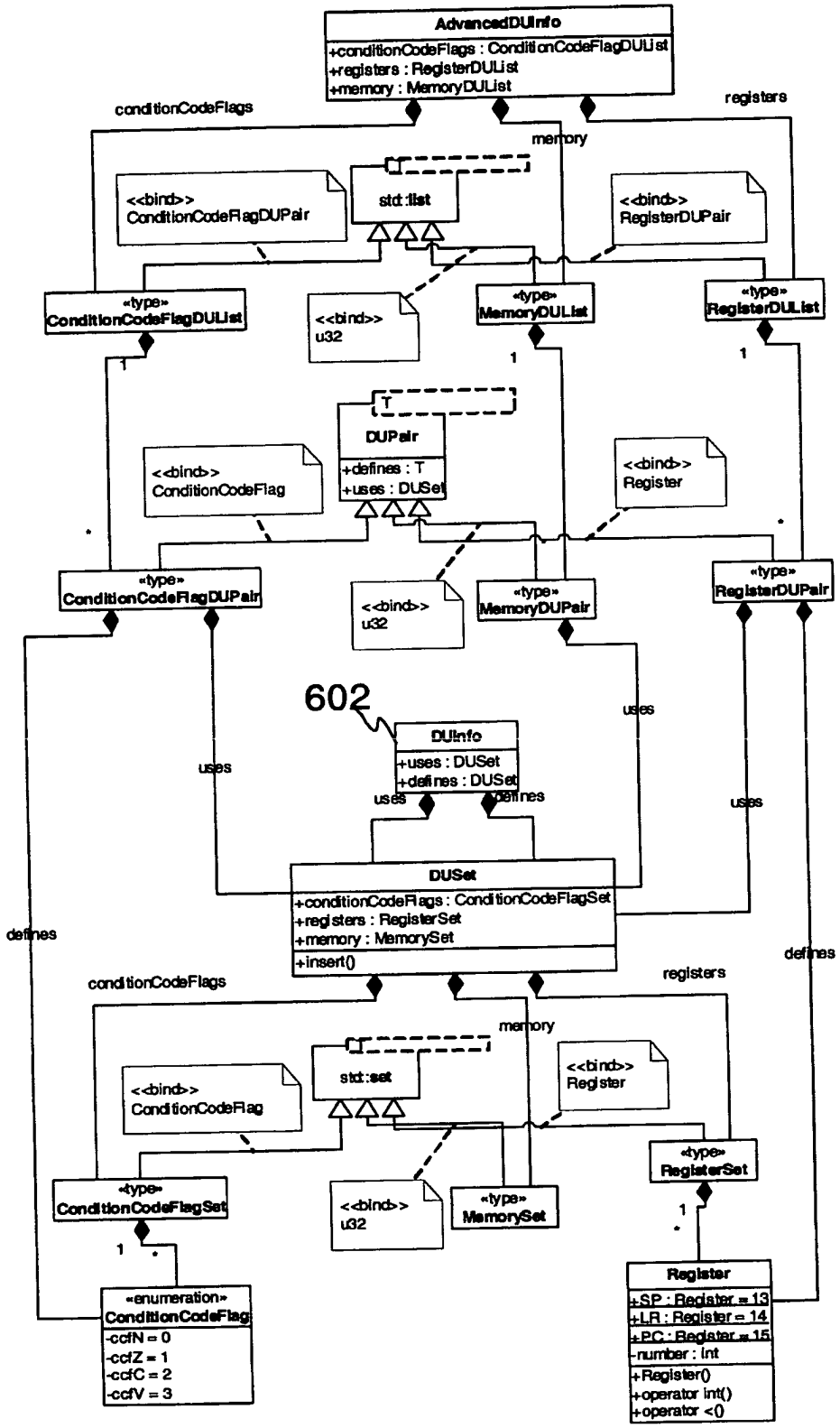

AdvancedDUInfo
+conditionCodeFlags : ConditionCodeFlagDUList
+registers : RegisterDUList
+memory : MemoryDUList
conditionCodeFlags
registers
memory
<<bind>>
ConditionCodeFlagDUPair
std::list
<<bind>>
RegisterDUPair
«type»
ConditionCodeFlagDUList
<<bind>>
u32
«type»
MemoryDUList
«type»
RegisterDUList
1
1
1
T
DUPair
+defines : T
+uses : DUSet
<<bind>>
ConditionCodeFlag
<<bind>>
Register
«type»
ConditionCodeFlagDUPair
<<bind>>
u32
«type»
MemoryDUPair
«type»
RegisterDUPair
602
DUInfo
+uses : DUSet
+defines : DUSet
uses
uses
uses
defines
uses
DUSet
+conditionCodeFlags : ConditionCodeFlagSet
+registers : RegisterSet
+memory : MemorySet
+insert()
defines
defines
conditionCodeFlags
registers
memory
<<bind>>
ConditionCodeFlag
std::set
<<bind>>
Register
«type»
ConditionCodeFlagSet
<<bind>>
u32
«type»
MemorySet
«type»
RegisterSet
1
1
*
«enumeration»
ConditionCodeFlag
-ccfN = 0
-ccfZ = 1
-ccfC = 2
-ccfV = 3
Register
+SP : Register = 13
+LR : Register = 14
+PC : Register = 15
-number : int
+Register()
+operator int()
+operator <()

Figure 6A

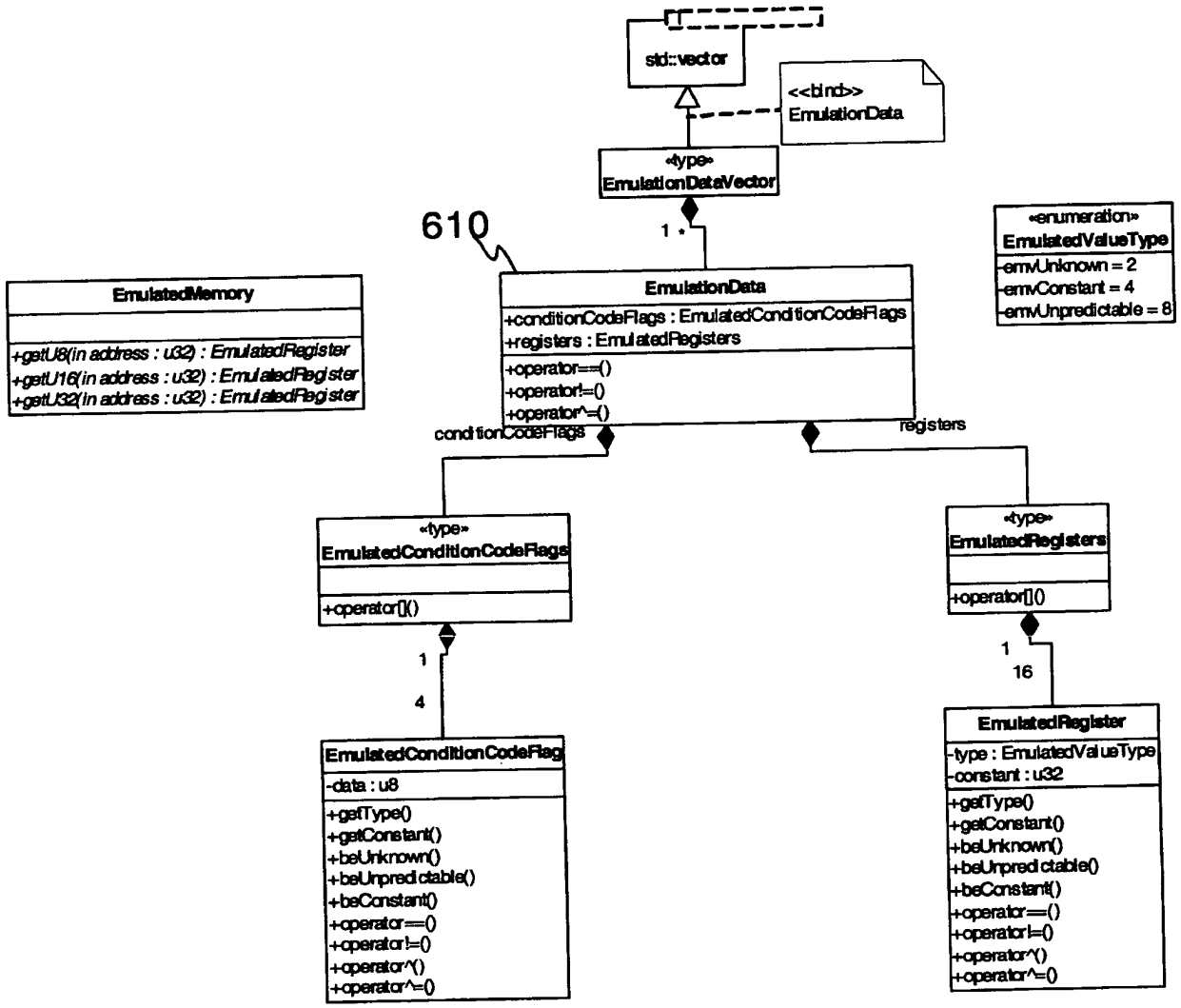

std::vector
<<bind>>
EmulationData
«type»
EmulationDataVector
610
1 *
«enumeration»
EmulatedValueType
-emvUnknown = 2
-emvConstant = 4
-emvUnpredictable = 8
EmulatedMemory
+getU8(in address : u32) : EmulatedRegister
+getU16(in address : u32) : EmulatedRegister
+getU32(in address : u32) : EmulatedRegister
EmulationData
+conditionCodeFlags : EmulatedConditionCodeFlags
+registers : EmulatedRegisters
+operator==()
+operator!=()
+operator^=()
conditionCodeFlags
registers
«type»
EmulatedConditionCodeFlags
+operator[]()
«type»
EmulatedRegisters
+operator[]()
1
4
1
16
EmulatedConditionCodeFlag
-data : u8
+getType()
+getConstant()
+beUnknown()
+beUnpredictable()
+beConstant()
+operator==()
+operator!=()
+operator^()
+operator^=()
EmulatedRegister
-type : EmulatedValueType
-constant : u32
+getType()
+getConstant()
+beUnknown()
+beUnpredictable()
+beConstant()
+operator==()
+operator!=()
+operator^()
+operator^=()

Figure 6B

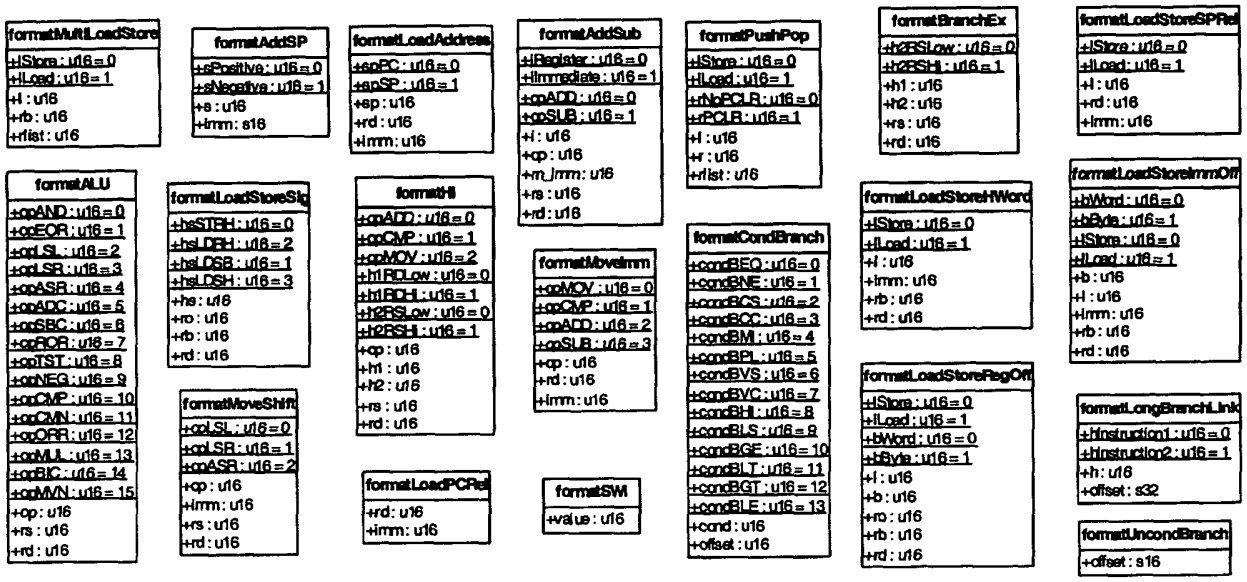


Figure 6C

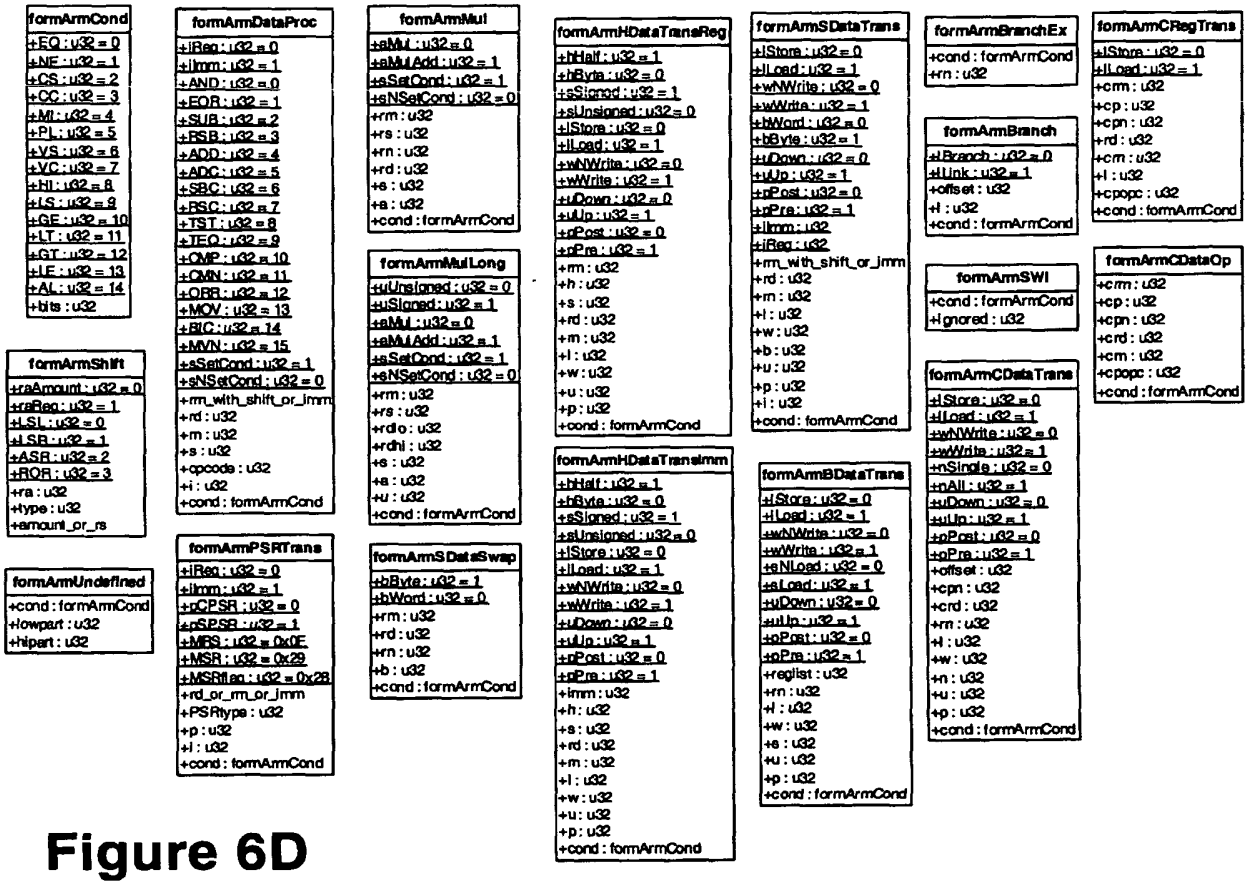
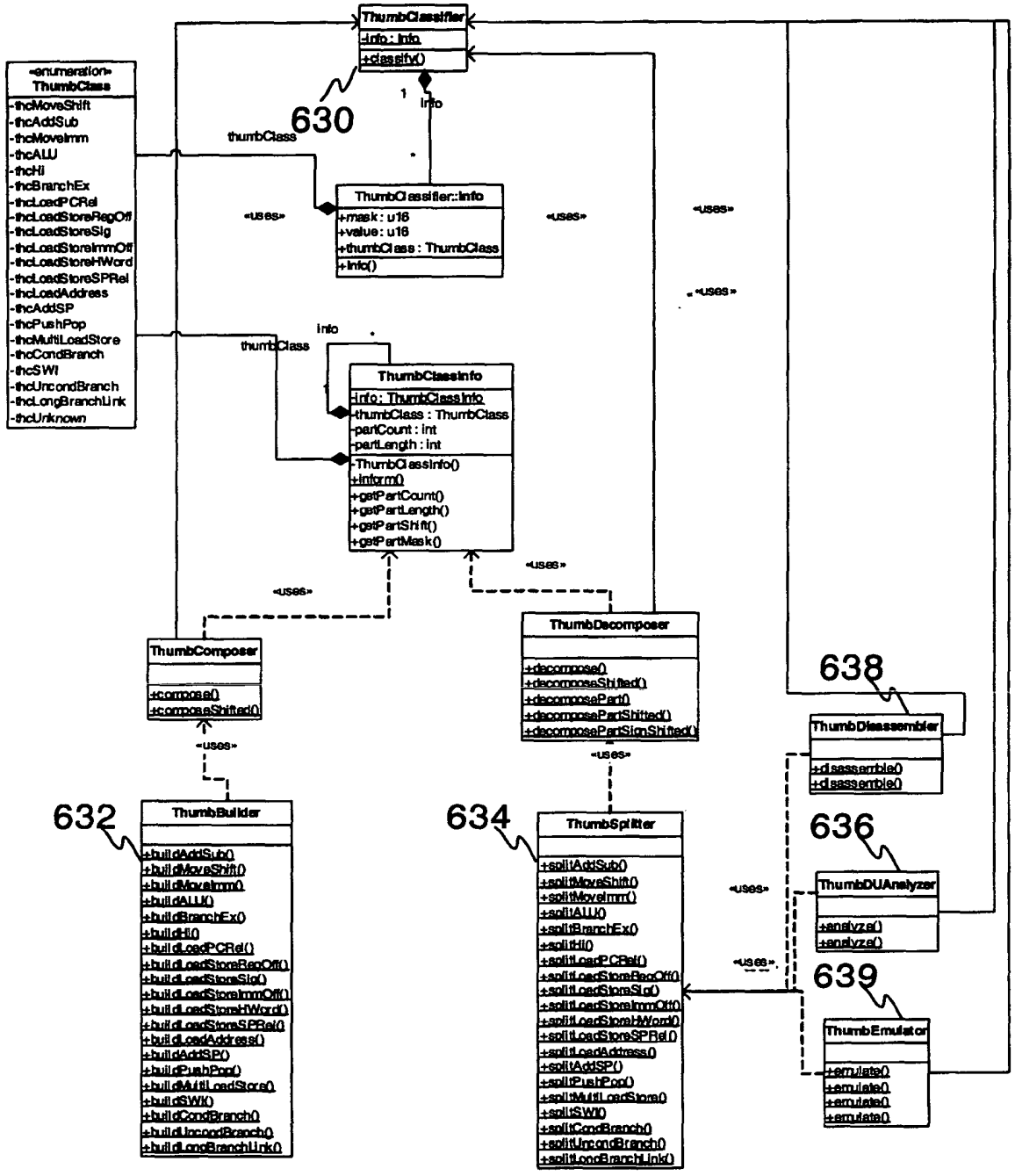


Figure 6D


ThumbClassifier
-info : Info
+classify()
630
1
Info
thumbClass
«enumeration»
ThumbClass
-thcMoveShift
-thcAddSub
-thcMoveImm
-thcALU
-thcHi
-thcBranchEx
-thcLoadPCRel
-thcLoadStoreRegOff
-thcLoadStoreSig
-thcLoadStoreImmOff
-thcLoadStoreHWord
-thcLoadStoreSPRel
-thcLoadAddress
-thcAddSP
-thcPushPop
-thcMultiLoadStore
-thcCondBranch
-thcSWI
-thcUncondBranch
-thcLongBranchLink
-thcUnknown
«uses»
ThumbClassifier::Info
+mask : u16
+value : u16
+thumbClass : ThumbClass
+Info()
«uses»
«uses»
«uses»
Info
thumbClass
ThumbClassInfo
-info : ThumbClassInfo
-thumbClass : ThumbClass
-partCount : int
-partLength : int
-ThumbClassInfo()
+inform()
+getPartCount()
+getPartLength()
+getPartShift()
+getPartMask()
«uses»
«uses»
ThumbComposer
+compose()
+composeShifted()
«uses»
ThumbDecomposer
+decompose()
+decomposeShifted()
+decomposePart()
+decomposePartShifted()
+decomposePartSignShifted()
«uses»
638
ThumbDisassembler
+disassemble()
+disassemble()
632
ThumbBuilder
+buildAddSub()
+buildMoveShift()
+buildMoveImm()
+buildALU()
+buildBranchEx()
+buildHi()
+buildLoadPCRel()
+buildLoadStoreRegOff()
+buildLoadStoreSig()
+buildLoadStoreImmOff()
+buildLoadStoreHWord()
+buildLoadStoreSPRel()
+buildLoadAddress()
+buildAddSP()
+buildPushPop()
+buildMultiLoadStore()
+buildSWI()
+buildCondBranch()
+buildUncondBranch()
+buildLongBranchLink()
634
ThumbSplitter
+splitAddSub()
+splitMoveShift()
+splitMoveImm()
+splitALU()
+splitBranchEx()
+splitHi()
+splitLoadPCRel()
+splitLoadStoreRegOff()
+splitLoadStoreSig()
+splitLoadStoreImmOff()
+splitLoadStoreHWord()
+splitLoadStoreSPRel()
+splitLoadAddress()
+splitAddSP()
+splitPushPop()
+splitMultiLoadStore()
+splitSWI()
+splitCondBranch()
+splitUncondBranch()
+splitLongBranchLink()
636
ThumbDUAnalyzer
+analyze()
+analyze()
«uses»
«uses»
639
ThumbEmulator
+emulate()
+emulate()
+emulate()
+emulate()

Figur 6E

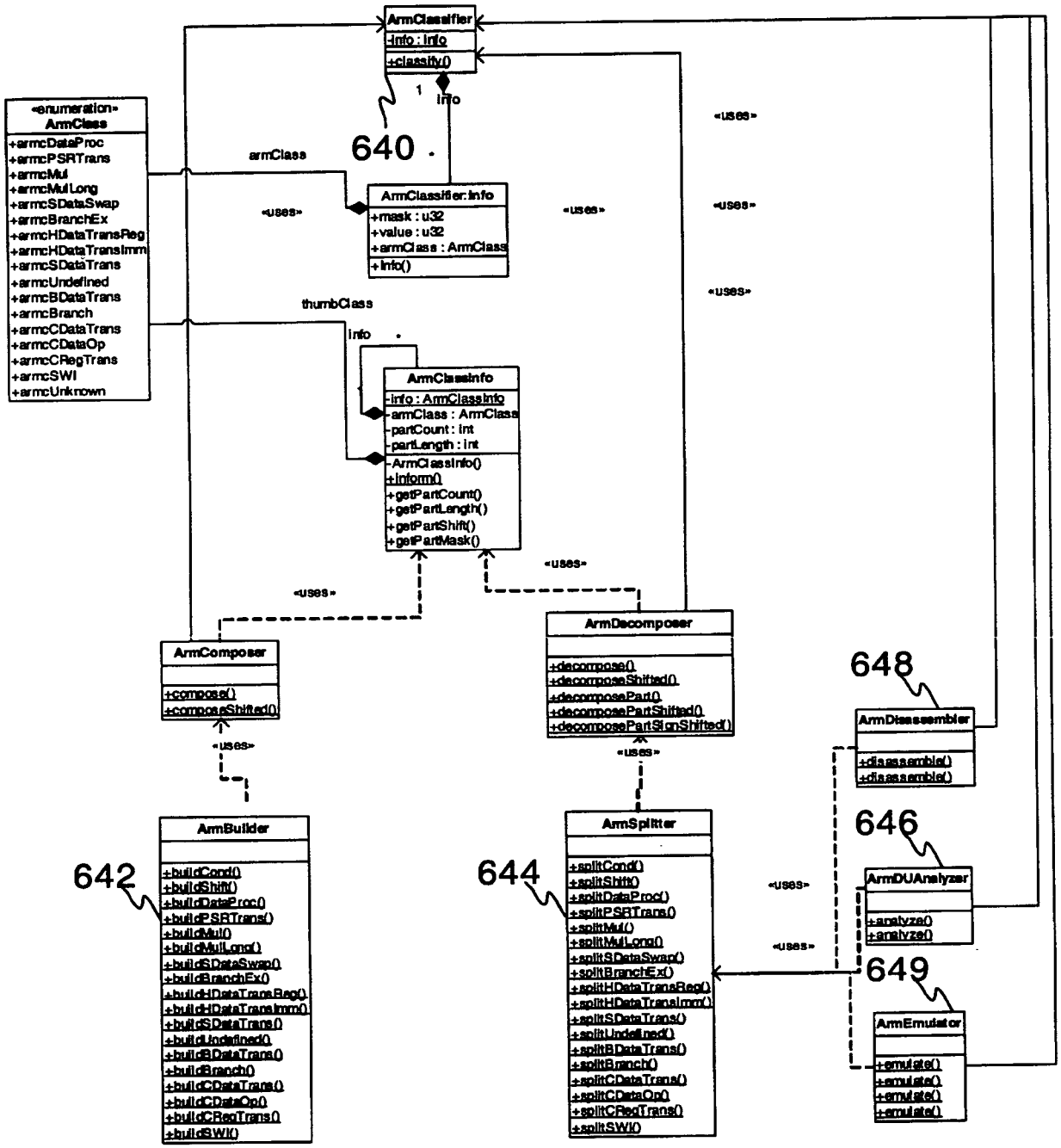

ArmClassifier
-Info : Info
+classify()
1
Info
640
armClass
«enumeration»
ArmClass
+armcDataProc
+armcPSRTrans
+armcMul
+armcMulLong
+armcSDataSwap
+armcBranchEx
+armcHDataTransReg
+armcHDataTransImm
+armcSDataTrans
+armcUndefined
+armcBDataTrans
+armcBranch
+armcCDataTrans
+armcCDataOp
+armcCRegTrans
+armcSWI
+armcUnknown
«uses»
ArmClassifier.Info
+mask : u32
+value : u32
+armClass : ArmClass
+Info()
thumbClass
info
ArmClassInfo
-info : ArmClassInfo
-armClass : ArmClass
-partCount : int
-partLength : int
-ArmClassInfo()
+inform()
+getPartCount()
+getPartLength()
+getPartShift()
+getPartMask()
ArmComposer
+compose()
+composeShifted()
ArmDecomposer
+decompose()
+decomposeShifted()
+decomposePart()
+decomposePartShifted()
+decomposePartSignShifted()
ArmBuilder
+buildCond()
+buildShift()
+buildDataProc()
+buildPSRTrans()
+buildMul()
+buildMulLong()
+buildSDataSwap()
+buildBranchEx()
+buildHDataTransReg()
+buildHDataTransImm()
+buildSDataTrans()
+buildUndefined()
+buildBDataTrans()
+buildBranch()
+buildCDataTrans()
+buildCDataOp()
+buildCRegTrans()
+buildSWI()
642
ArmSplitter
+splitCond()
+splitShift()
+splitDataProc()
+splitPSRTrans()
+splitMul()
+splitMulLong()
+splitSDataSwap()
+splitBranchEx()
+splitHDataTransReg()
+splitHDataTransImm()
+splitSDataTrans()
+splitUndefined()
+splitBDataTrans()
+splitBranch()
+splitCDataTrans()
+splitCDataOp()
+splitCRegTrans()
+splitSWI()
644
ArmDisassembler
+disassemble()
+disassemble()
648
ArmDUAnalyzer
+analyze()
+analyze()
646
ArmEmulator
+emulate()
+emulate()
+emulate()
+emulate()
649

Figure 6F

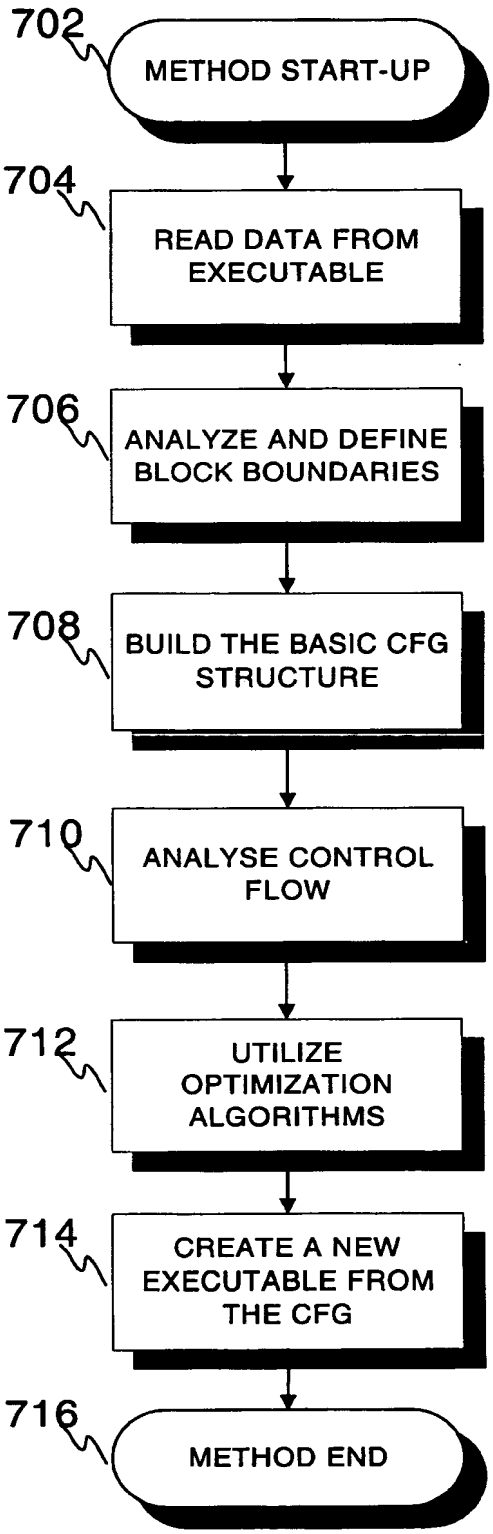

702
METHOD START-UP
704
READ DATA FROM EXECUTABLE
706
ANALYZE AND DEFINE BLOCK BOUNDARIES
708
BUILD THE BASIC CFG STRUCTURE
710
ANALYSE CONTROL FLOW
712
UTILIZE OPTIMIZATION ALGORITHMS
714
CREATE A NEW EXECUTABLE FROM THE CFG
716
METHOD END

Figure 7A

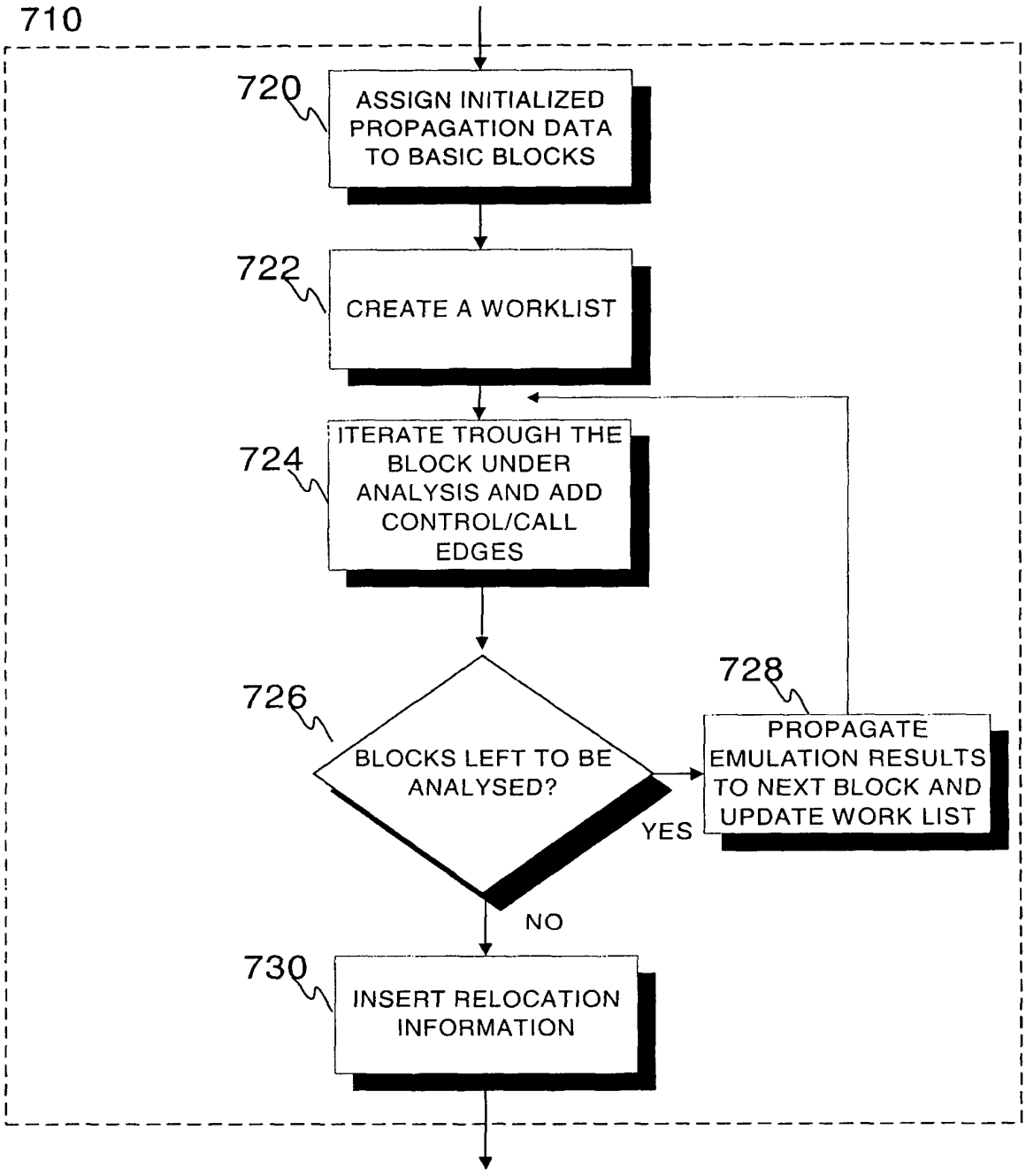

710
720
ASSIGN INITIALIZED PROPAGATION DATA TO BASIC BLOCKS
722
CREATE A WORKLIST
724
ITERATE TROUGH THE BLOCK UNDER ANALYSIS AND ADD CONTROL/CALL EDGES
726
BLOCKS LEFT TO BE ANALYSED?
YES
NO
728
PROPAGATE EMULATION RESULTS TO NEXT BLOCK AND UPDATE WORK LIST
730
INSERT RELOCATION INFORMATION

Figure 7B